IN-32
348965

SEMI-ANNUAL STATUS REPORT

Error Control Coding Techniques for Space and Satellite Communications

NASA Grant Number NAG5-557
Report Number 98-001

Principal Investigator: Daniel J. Costello, Jr.
Post-Doctoral Associate: Oscar Y. Takeshita
Graduate Assistant: Hermano A. Cabral
Department of Electrical Engineering
University of Notre Dame

August 1, 1998

Department of
ELECTRICAL ENGINEERING

UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

Part I

Asymmetric Turbo-Codes

Oscar Y. Takeshita and Daniel J. Costello, Jr.

University of Notre Dame

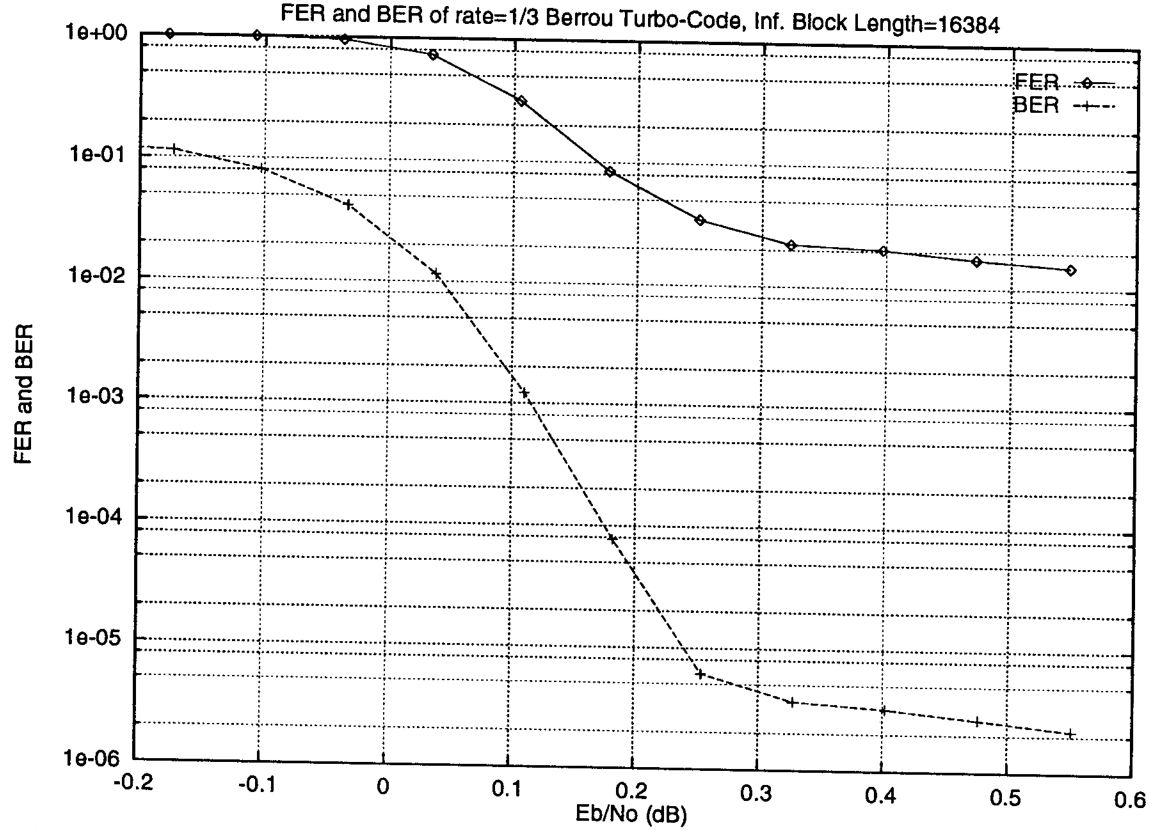
Outline

- Introduction
- BER vs. FER for the Berrou Code
- The FER of "Primitive" and "Asymmetric" Turbo-Codes
- Pathological Frames generated by the Iterative MAP Decoder
- Improving the FER and the BER with an Outer Code
  - Outer Code Across Frames
  - Outer Code within Each Frame
- Conclusions

Introduction

- It is well known that the BER performance of a parallel concatenated turbo-code improves roughly as $1/N$, where N is the information block length. However, it has been observed by Benedetto and Montorsi that for most parallel concatenated turbo-codes, the FER performance does not improve monotonically with N.
- In this report, we study the FER of turbo-codes, and the effects of their concatenation with an outer code. Two methods of concatenation are investigated: across several frames and within each frame. Some asymmetric codes are shown to have excellent FER performance with an information block length of 16384.
- We also show that the proposed outer coding schemes can improve the BER performance as well by eliminating pathological frames generated by the iterative MAP decoding process.

BER vs. FER for the Berrou Code



In spite of the excellent BER performance of the rate 1/3, 16-state Berrou code with an information block length of 16384, its FER performance is very poor. This can be attributed to its small minimum distance, which also causes the "error-floor" phenomenon.

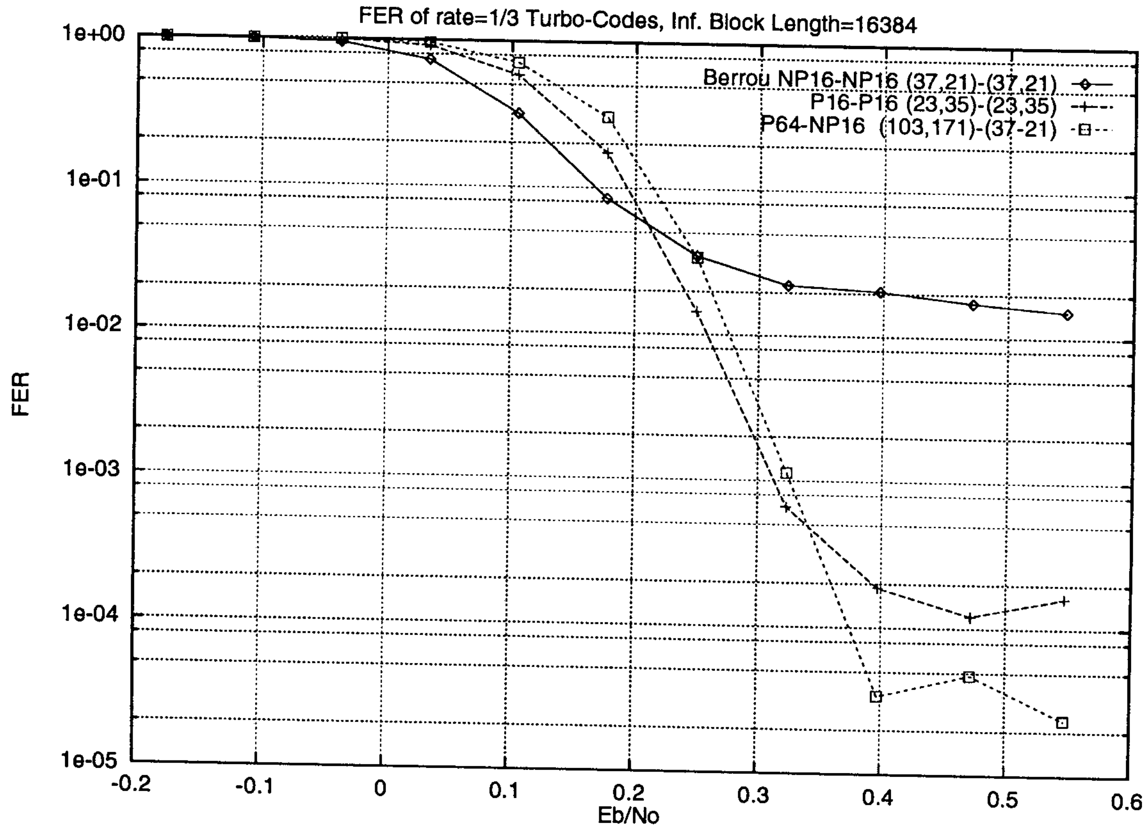
Asymmetric Turbo-Codes

Asymmetric turbo-codes are parallel concatenated turbo-codes with different component codes. The following component codes appear in this report:

Table 1: Table of Component Codes

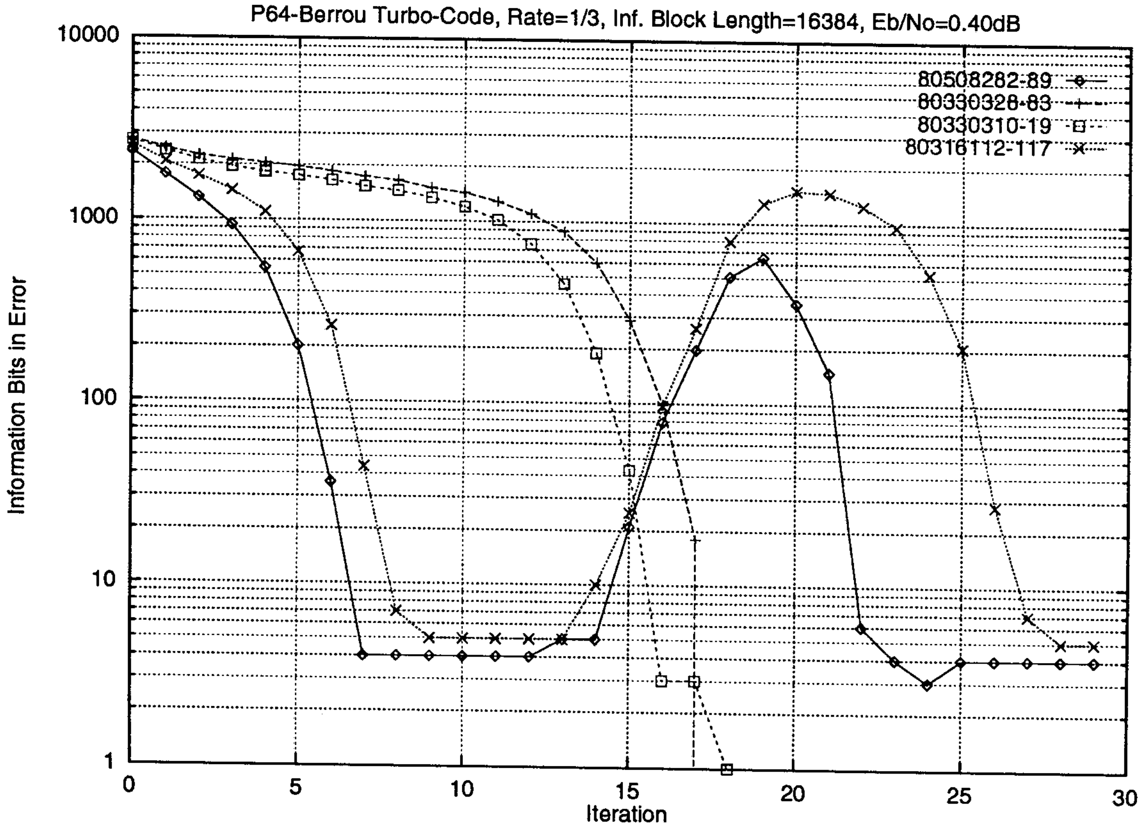
Code	Feedback poly (oct)	Feedforward poly (oct)
Berrou	37 (non-primitive)	21
P16	23 (primitive)	35
P32	51 (primitive)	77
P64	103 (primitive)	171

The FER of "Primitive" and "Asymmetric" Turbo-Codes



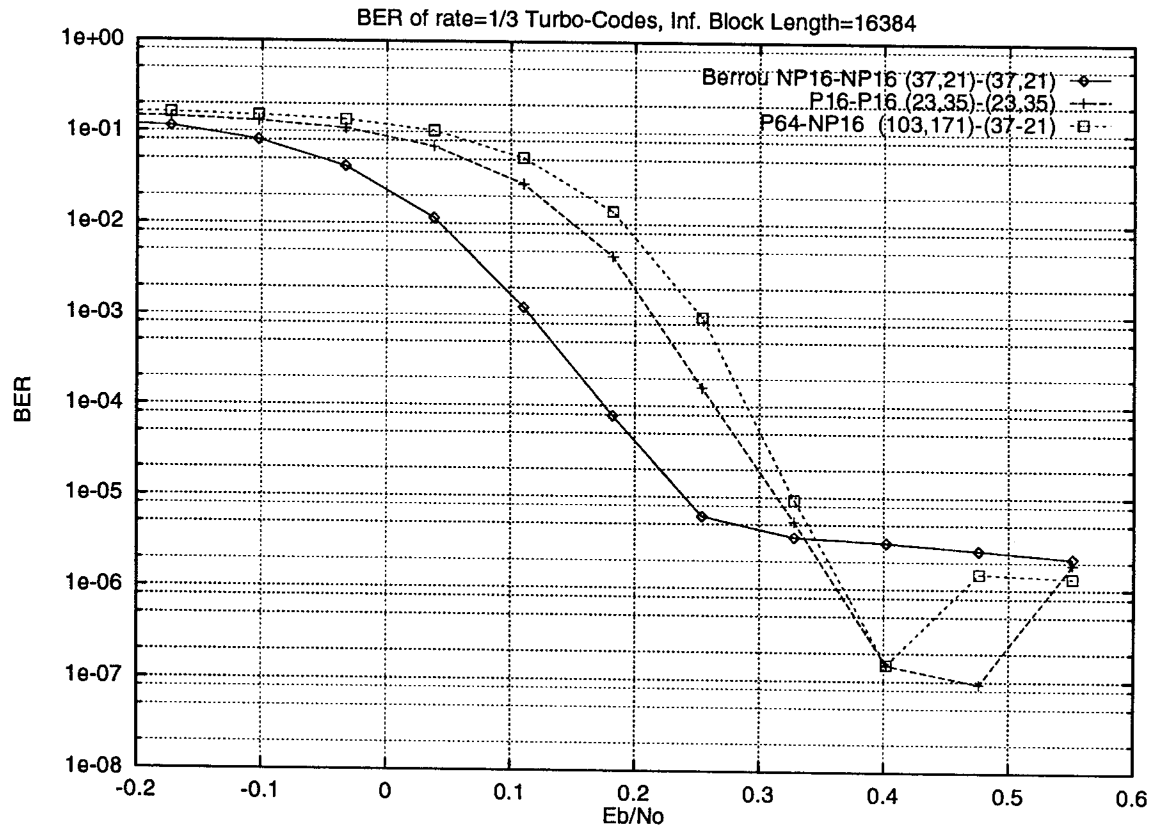
"Primitive" and "asymmetric (ISIT'98)" turbo-codes, i.e., parallel concatenated turbo-codes with different component codes, give good FER performance.

Pathological Frames generated by the Iterative MAP Decoder



A pathological frame: 1) takes much longer than the average number of iterative MAP decoding iterations to converge, 2) oscillates, or 3) does not converge.

The Effect of Pathological Frames on the BER



Pathological frames influence the BER curve in an odd way if a fixed number of iterations (18 in this case) is used.

Outer Coding Across Frames (1)

We have studied a scheme to improve the FER and BER performance of "primitive" and "asymmetric" turbo-codes.

Typical behavior of a "primitive" or "asymmetric" turbo-code:

1. The FER of the turbo-code is below 10^{-4}.
2. The bit error locations in an erroneous frame are not random.
3. There is more than 1 bit error in each erroneous frame.

We can improve the FER and BER performance of the turbo-code by using:

1. A (127,120) Hamming code across a block of 127 frames.
2. A different scrambler for each frame prior to turbo encoding.
3. A simple algorithm to correct up to 2 erroneous frames in a block.

Outer Coding Across Frames (2)

In the next three slides we show figures corresponding to typical cases of the decoding algorithm. First we sort the rows containing errors corrected by the Hamming outer code in decreasing order of the number of errors corrected per row. Then we have the following cases:

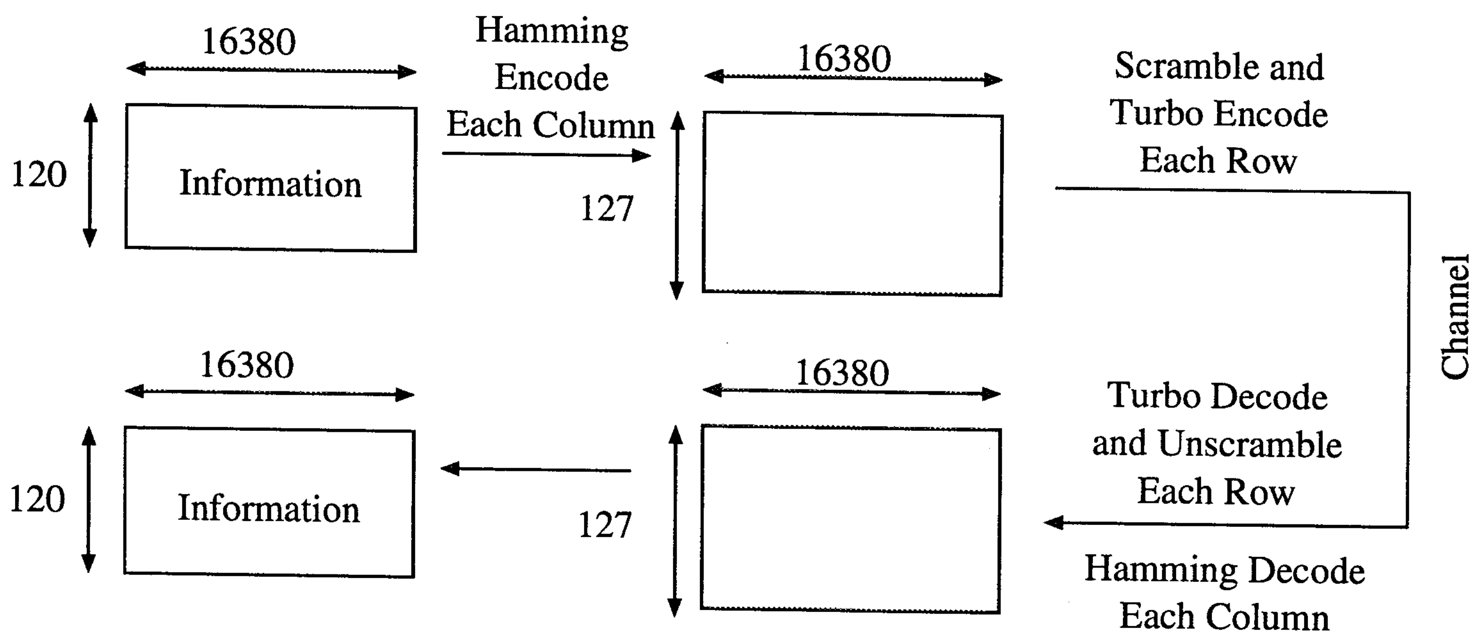
- Only one row contains errors corrected by the Hamming code. In this case, we take no further action and assume decoding is correct.

- Two rows contain errors corrected by the Hamming code. (The corrected positions must be disjoint.) Again, we take no further action and assume decoding is correct.

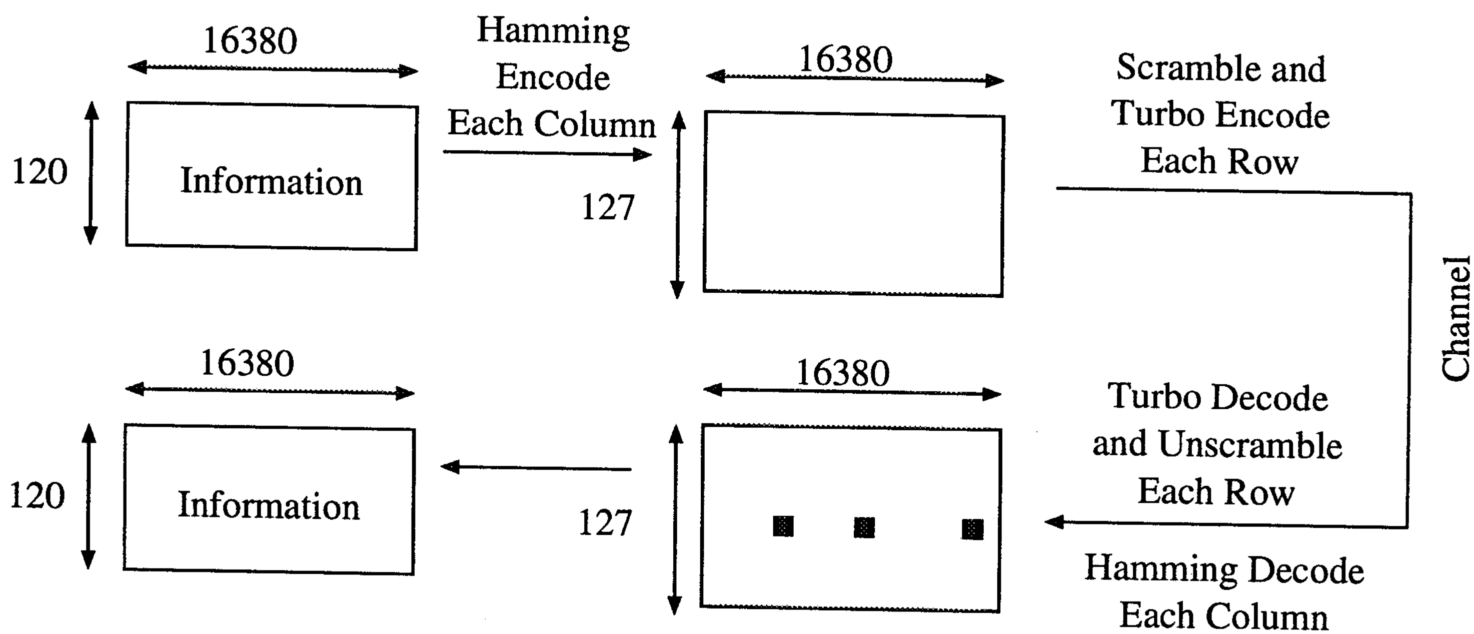
- Three rows contain errors corrected by the Hamming code, but the rows are linearly dependent. In this case, we assume that the row with the smallest number of errors was actually correct, and instead, errors in the corresponding positions occured in the other two rows.

If three or more rows contain errors corrected by the Hamming code, and the rows are not linearly dependent, we cancel outer decoding and take the result of the inner turbo-code, since the probability of miscorrection is high.

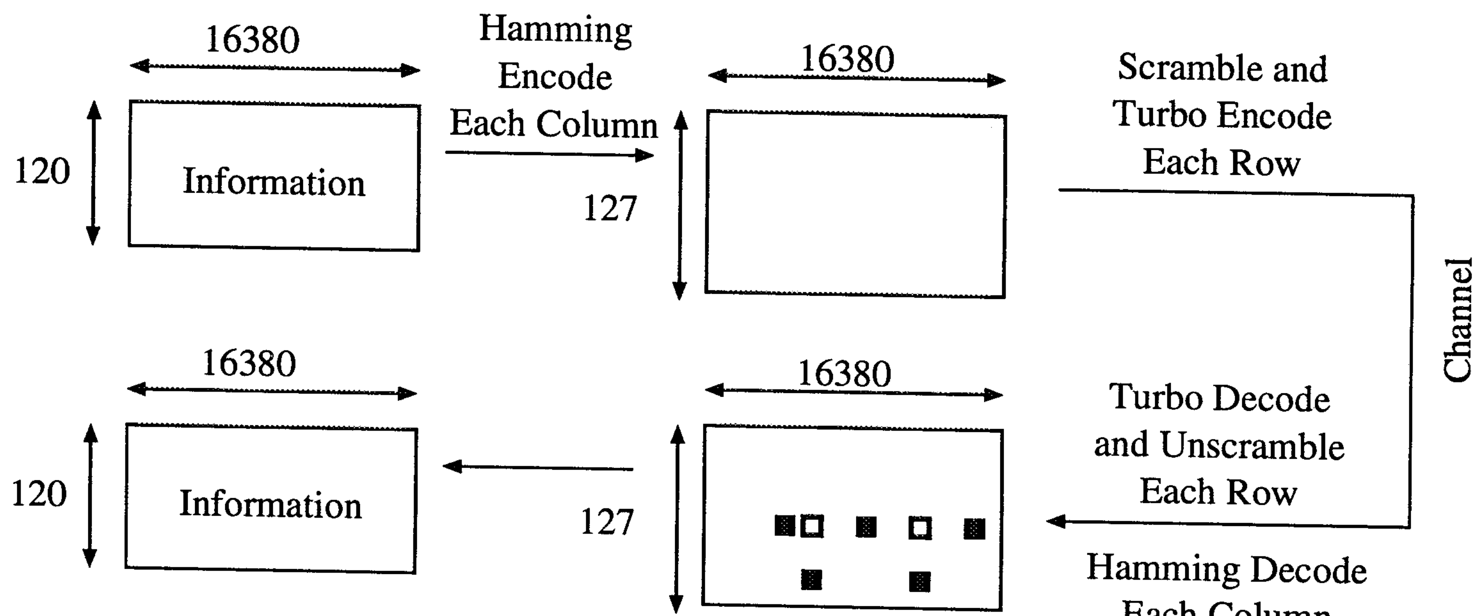
Outer Coding Across Frames (3)



Outer Coding Across Frames (4)



Outer Coding Across Frames (5)



Row with the largest number of errors

Row with the second largest number of errors

Row with the third largest number of errors
(This row is a linear combination of and)

Assume these positions are correct

Assume these positions are in error

Outer Coding Across Frames (6)

- For a FER of $f_e^i = 10^{-4}$ after inner decoding, the probability of no frames, one frame, and two frames in error in a block of 127 frames are $p(0) = (1 - f_e^i)^{127}$, $p(1) = 127 f_e^i (1 - f_e^i)^{126}$, and $p(2) = 8001 (f_e^i)^2 (1 - f_e^i)^{125}$, respectively.

- Assuming that the case with more than two frames in error at the outer decoder is not incorrectly decoded as one or two frame errors (a very unlikely event), the FER after outer decoding can be estimated as $f_e^o = f_e^i - p(1)/127 - 2p(2)/127 = 7.8 \times 10^{-9}$, a reduction of about four orders of magnitude.

- Moreover, because of the pathological frames generated by the iterative MAP decoder, the BER of the concatenated scheme is also greatly improved.

Outer Coding within Each Frame (1)

- Simulation results indicate that even at low SNR's, the FER performance of asymmetric turbo-codes can be greatly improved by allowing a larger maximum number of iterations and using an outer BCH code with a correction capability of 5 errors on top of each frame. For a code length of 16384 information bits and a rate of 1/3, the rate loss due to the BCH code is only 0.03dB.

- This improvement is possible because for asymmetric codes, almost all frames in error result in a small number of information bits in error even at low SNR's. Moreover, the BCH code provide a very reliable (small undetected error probability) stopping criterion for the iterations. We also note that the FER performance of the symmetric turbo-codes are greatly improved by the same technique.

Outer Coding within Each Frame (2)

- The undetected error probability of a BCH code is calculated assuming that each of the outputs of the iterative MAP decoder during the convergence process are uncorrelated with the codewords of the BCH code. For a BCH code with parameters (n, k, t, τ), where n is the codeword length, k is the information length, t is the maximum error correcting capability of the code, and $\tau =< t$ is the actual number of errors corrected by the decoder, the undetected error probability p_u can be estimated as:

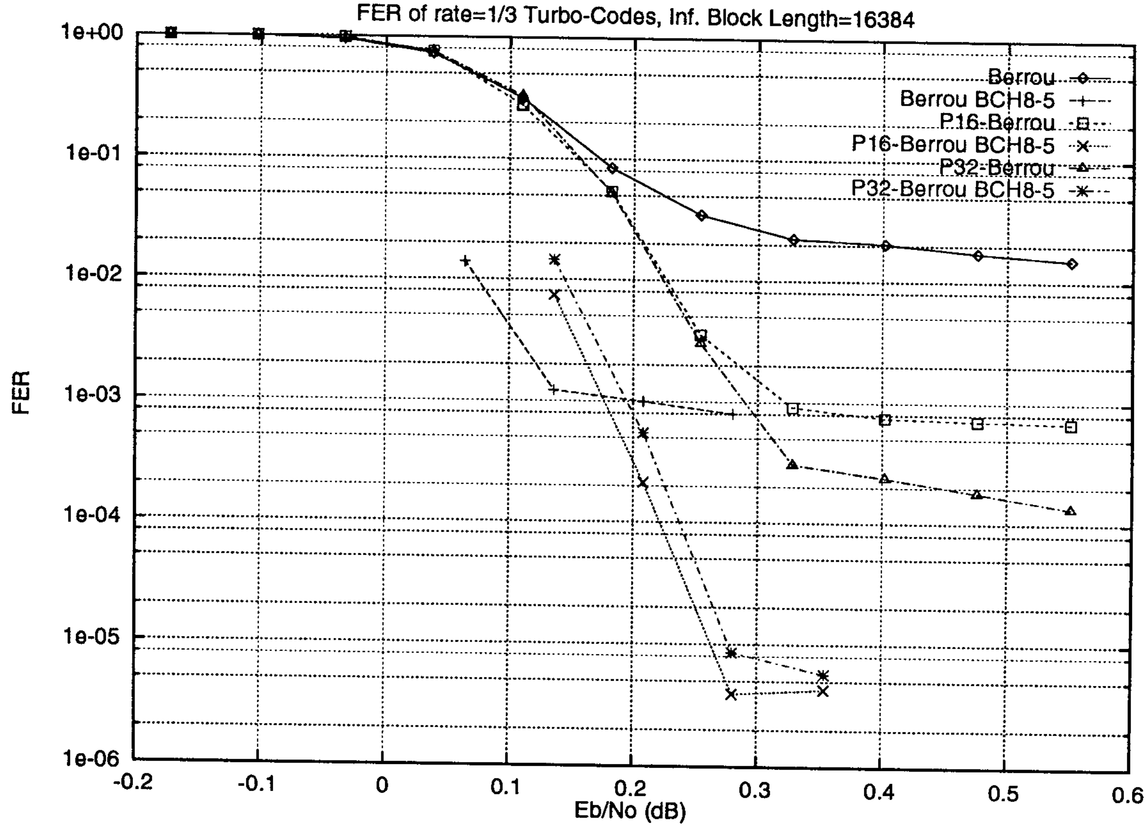
$$p_u = \frac{(2^k - 1)\binom{n}{\tau}}{2^n} \approx \frac{\binom{n}{\tau}}{2^{n-k}}.$$

- Note that because of the size of the parameters that we intend to use for the outer code (large n and k, small t and τ), the undetected error probability p_u can be drastically reduced by making τ slightly smaller than t. For example, for a $(16383, 16285, 7, 7)$ code, $p_u = 1.98 \times 10^{-4}$, whereas for a $(16383, 16285, 7, 6)$ code, $p_u = 8.47 \times 10^{-8}$.

Outer Coding within Each Frame (3)

In the next slide we show the FER performance of an asymmetric P16-Berrou code, an asymmetric P32-Berrou code, and the symmetric Berrou code with a (16383,16271,8,5) BCH outer code. The number of redundant bits $n - k$ of this BCH code is 112, and hence $p_u = 1.89 \times 10^{-15}$. A maximum of 50 iterations is allowed. Note, however, that for the simulated SNR range, the average number of iterations per frame is much lower (on the order of 10) than the maximum due to the stopping criterion provided by the BCH code. An implementation of the turbo codec requires 16380 information bits and 4 tail bits for a 16-state component code. The outer code is applied only to the information bits and then the tail bits are computed. Hence we use a shortened (16380,16268,8,5) BCH outer code.

Outer Coding within Each Frame (4)



Conclusions (1)

- The P64-NP16 turbo-code provides very good FER performance of about 5×10^{-5} at an SNR of 0.4dB, which is more than 2 orders of magnitude better than the FER performance of the 16-state Berrou code at the same SNR.
- The concatenation of the P64-NP16 turbo-code with a Hamming code across 127 frames provides a coding scheme with an FER on the order of 10^{-8}, although decoding delay is increased.
- Also, the concatenation greatly improves the BER performance by eliminating the large number of bit errors in the iterative MAP decoder due to pathological frames.

Conclusions (2)

- BCH outer coding within each frame is a very robust coding scheme. It provides an excellent stopping criterion for the iterations of the MAP decoder and also efficiently corrects some of the pathological frames.

- The undetected error probability of the BCH code can be made very small by reducing its error correction capability.

- Simulation results show an impressively low FER of 10^{-5} at an SNR of only 0.26dB, an improvement of more than three orders of magnitude compared to turbo decoding without an outer code, for a rate 1/3, information block length 16384 turbo-code. This method also eliminates the long decoding delay resulting from using an outer code across several frames.

- The rate loss due to outer BCH coding within each frame is only approximately 0.03dB.

Part II

Multiple Parity Sequence Bootstrap Hybrid Decoding

Hermano A. Cabral and Daniel J. Costello, Jr.

University of Notre Dame

Introduction

- Sequential Decoding has long been known to almost optimally exploit the full error-correcting capabilities of a large constraint length convolutional code. Its disadvantage is that it takes a variable amount of computation to finish decoding, with an infinite average number of computations for code rates beyond a given value, called the computational cut-off rate R_0, which is strictly less than the channel capacity.

- In 1971, an iterative coding/decoding scheme, called *Bootstrap Hybrid Decoding* (BHD), was proposed that uses a sequential decoder, but alleviates some of its problems through the use of an additional parity constraint. The performance gain in terms of E_b/N_0 over a normal sequential decoder is around 1.0dB.
- This report discusses an extension of the BHD scheme to multiple parity sequences and compares the performance obtained with that of the original scheme.

Sequential Decoding

- It is well known that large constraint length convolutional codes exist with very good error-correcting capabilities, which when decoded by an optimum decoder yield very low Bit Error Rates (BER's) at rates arbitrarily close to capacity. Unfortunately, optimum decoders for these codes are not practical to implement due to their large complexity.
- Sequential decoding, on the other hand, is a suboptimum decoding algorithm whose complexity is almost independent of the constraint length. Its suboptimum performance can be overcome by using a slightly larger constraint length.

- The disadvantage of sequential decoding is its probabilistic behavior, since its computational load depends on the noise and is therefore a random variable. In particular, for rates above the computational cut-off rate R_0, its computational load has an infinite expected value, and hence R_0 is considered the practical limit for sequential decoding.

The Bootstrap Hybrid Decoding Idea

- The BHD scheme works, as shown in the next figure, by adding an extra parity constraint to a set of m codewords, each L bits long, from a convolutional code, resulting in an additional sequence which, due to linearity, is a codeword itself.

- Here $\mathbf{x}_j = (x_{j,1}, \cdots, x_{j,L})$ is the j-th codeword, for $j = 1, \cdots, m$, and $\mathbf{x}_{m+1} = (x_{m+1,1}, \cdots, x_{m+1,L})$ is the additional parity sequence. We will also denote the i-th column in the above array by $\mathbf{c}_i^T$, i.e., $\mathbf{c}_i = (x_{1,i}, x_{2,i}, \cdots, x_{m+1,i})$, $i = 1, \cdots, L$.

The Bootstrap Hybrid Decoding Idea

$$
\begin{array}{cccccccc}
\boxed{x_{1,1}} & \boxed{x_{1,2}} & \boxed{x_{1,3}} & \cdots & \boxed{x_{1,L-1}} & \boxed{x_{1,L}} & \\
\boxed{x_{2,1}} & \boxed{x_{2,2}} & \boxed{x_{2,3}} & \cdots & \boxed{x_{2,L-1}} & \boxed{x_{2,L}} & \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \\
\boxed{x_{m,1}} & \boxed{x_{m,2}} & \boxed{x_{m,3}} & \cdots & \boxed{x_{m,L-1}} & \boxed{x_{m,L}} & \oplus \\
\hline
\boxed{x_{m+1,1}} & \boxed{x_{m+1,2}} & \boxed{x_{m+1,3}} & \cdots & \boxed{x_{m+1,L-1}} & \boxed{x_{m+1,L}} &
\end{array}
$$

- All codewords are transmitted over a noisy channel. The sequential decoder then uses the parity constraint to improve its likelihood function, thereby improving the effective computational cut-off rate R_0.
- The decoding scheme becomes iterative by using the estimated sequences provided by the sequential decoder to update the likelihood function, since, with high probability, these estimates are correct.

Multi-parity-sequence BHD

- The original BHD scheme made use of only one parity sequence, which limited its performance. Further gains can be achieved by using multiple parity sequences which check subsets of the set of codewords.
- In this case, the likelihood function adjustment is based on a set of values derived from the received values and the added parity constraints. For general parity constraints, the cardinality of this set is roughly exponential in the number of constraints, which limits the scheme to a few parity sequences in practice.

The Likelihood Function

- To derive the likelihood adjustment, assume that there are originally m codewords, each L bits long, and that l parity sequences are created from them, based on a linear code C, in a similar way as the original BHD. Also assume that

$$H = (\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_m, \mathbf{e}_1, \cdots, \mathbf{e}_l)$$

is the parity check matrix (in systematic form) of the code C, where $\mathbf{h}_i$, $i = 1, \cdots, m$, and $\mathbf{e}_j$, $j = 1, \cdots, l$, are column vectors of size l, and the vectors $\mathbf{e}_j$ consist of a single 1 in the j-th position.

- We now assume that k sequences have been successfully decoded, and hence that $m' = m + l - k$ sequences remain to be decoded. Then, by partitioning the set of $m + l$ sequences into two subsets, D, containing the k decoded sequences, and $\bar{D}$, its complement, the parity constraints imply that the sum of the sequences in the first subset is equal to the sum of the sequences in the second subset.

- The likelihood function to be used by the sequential decoder at time i to decode the j-th sequence is,

$$\lambda_i(j) = \log \frac{p(\mathbf{y}_i \mid x_{j,i}, \mathbf{c}_{i,D})}{p(\mathbf{y}_i \mid \mathbf{c}_{i,D})} - R,$$

where R is the code rate in bits per channel use, $\mathbf{y}_i$ is the received vector corresponding to $\mathbf{c}_i$, and $\mathbf{c}_{i,D}$ is the vector obtained by keeping the components of $\mathbf{c}_i$ corresponding to the sequences in D.

- After some manipulation, the likelihood function can be rewritten as

$$\lambda_i(j) = \log \frac{p(y_{j,i} \mid x_{j,i})}{p(y_{j,i})} - R + \hat{\lambda}_i(\mathbf{y}_{i,\bar{D}}, \bar{D}),$$

where $\mathbf{y}_{i,\bar{D}}$ is the vector containing the components of $\mathbf{y}_i$ corresponding to the sequences in $\bar{D}$, and $\hat{\lambda}_i(\mathbf{y}_{i,\bar{D}}, \bar{D})$ is the adjustment term due to the parity constraints. This term can be computed directly by adding $2^l - 1$ values, called channel state values, which are computed upon receiving the codeword and are updated during the decoding process.

The Iterative Procedure

- If the j-th sequence is successfully decoded, then the channel state sequences are updated (basically a division operation for each value in each sequence) based on the decoding estimate for that sequence, which is assumed correct and therefore determines the channel noise for that sequence. Note that it is important to have a highly reliable estimate, or else the channel state update would be incorrect, which could cause error propagation. For this reason, a sequential decoder with a long constraint length code is preferred.

- The decoder then checks if there is any undecoded sequence that can be decoded based on the parity constraints. This is done based on the column vectors $\hat{\mathbf{c}}_i$, which contain the decoder estimates for $\mathbf{c}_i$. For sequences in $\bar{D}$, the corresponding components of $\hat{\mathbf{c}}_i$ will be blank, i.e., treated as erasures, and therefore can be decoded (successfully or not) based on the erasure-correcting capabilities of the column code C.

- The decoder then proceeds to decode the next undecoded sequence with the updated channel state sequences, until no more undecoded erasures remain or some maximum number of computations is exceeded, in which case the overall frame is declared an erasure and the decoded sequences in D are released.

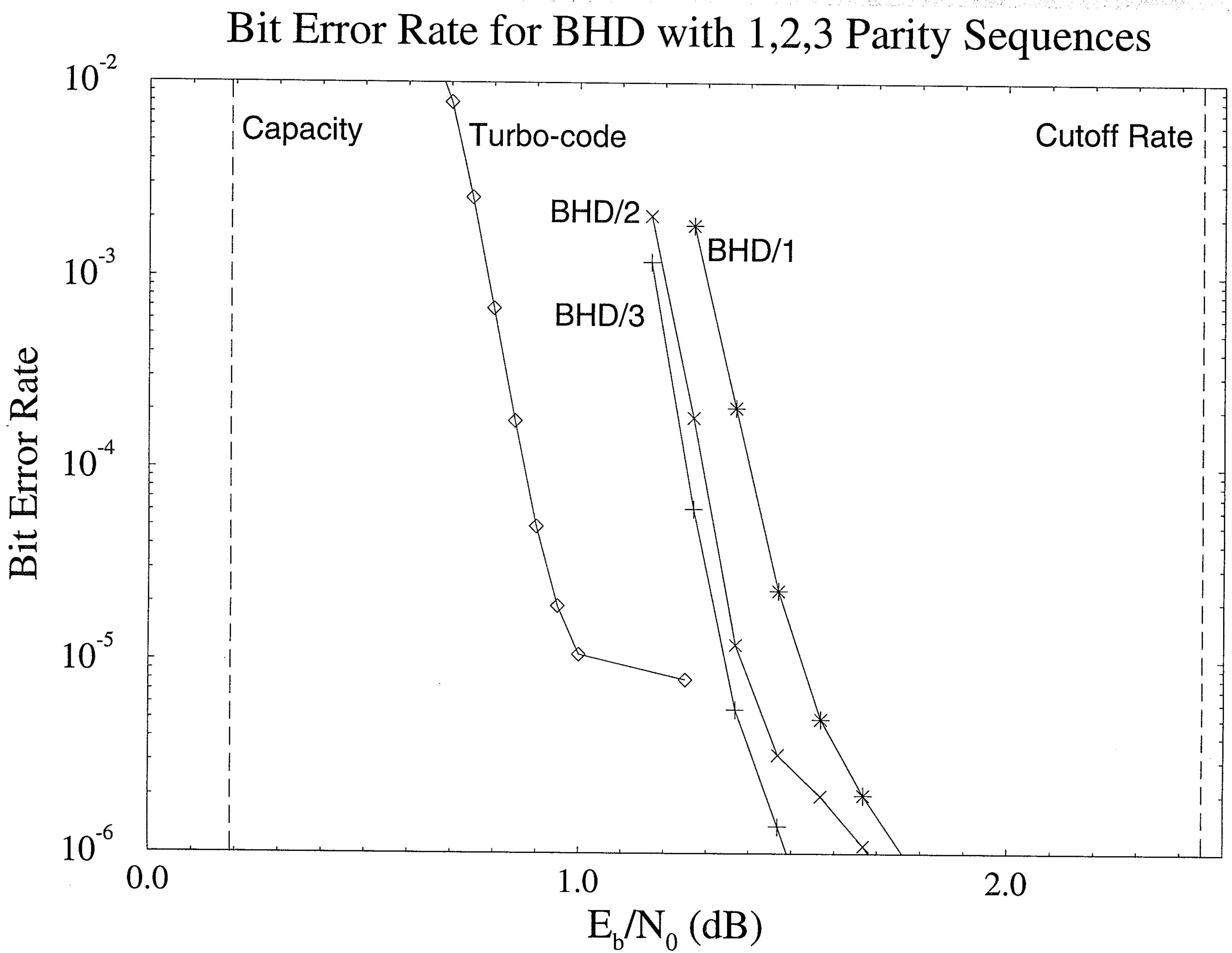
Simulation Results

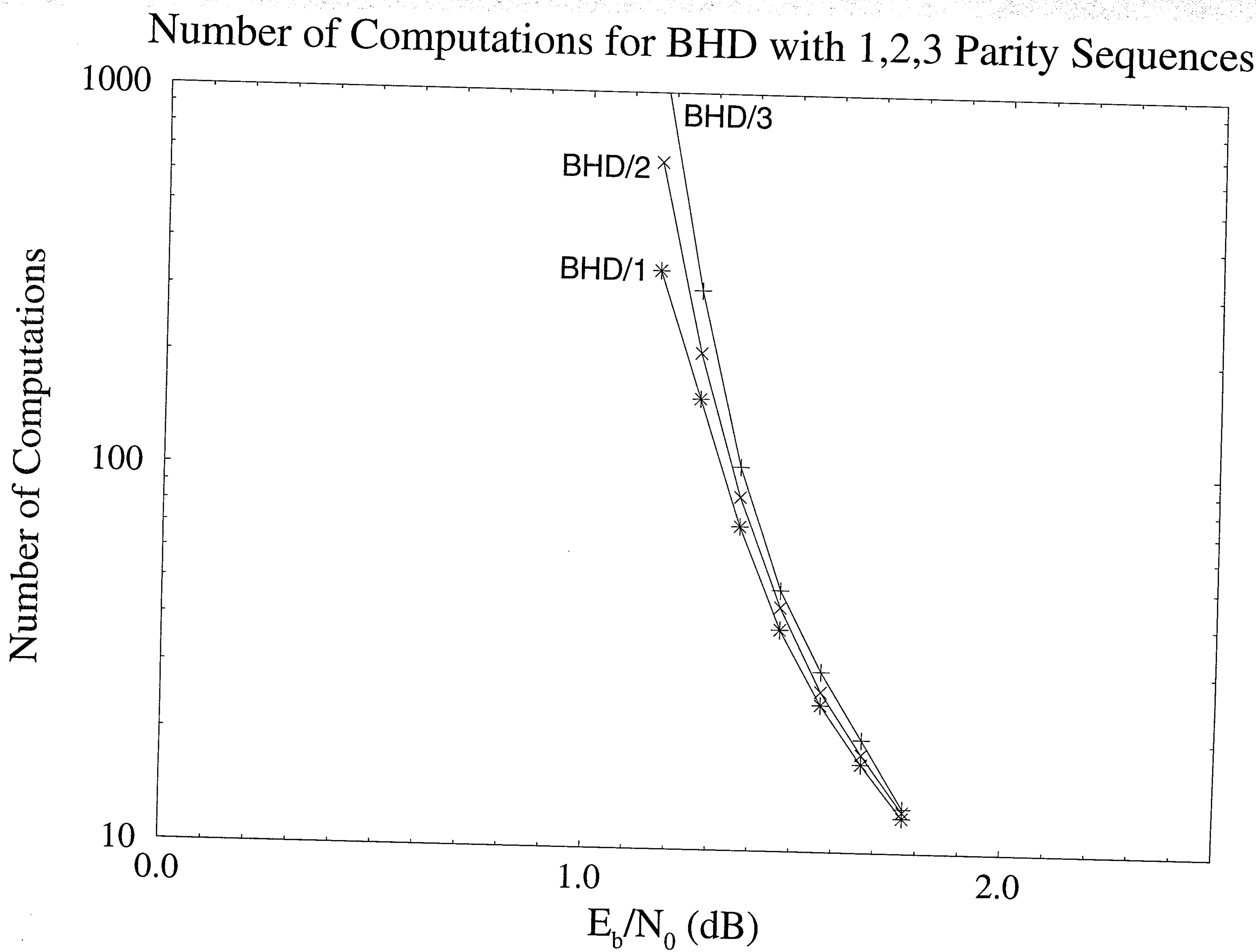
- Computer simulations have been performed to evaluate the performance of the multiple parity sequence BHD. Two cases were selected, one with two parity sequences and 18 codewords, and the other with three parity sequences and 27 codewords. In both cases each codeword contains 1000 information bits. Results for these two cases are shown in the next two figures and are labeled BHD/2 and BHD/3, respectively.

- Also shown is the single parity sequence BHD (label BHD/1) and a turbo-code, both for a block length of 9000 information bits.

- It can be seen that the two parity sequence BHD has a 0.15dB advantage at a BER of 10^{-5} compared to the original BHD scheme. The three parity sequence BHD gains another 0.05dB.
- Turbo-codes perform about 0.5dB better than the BHD schemes at BER's above 10^{-5}, but they suffer from an "error floor" at BER's below 10^{-5}.

- All three BHD schemes perform about the same number of computations per decoded bit, although this is skewed by the fact that the multiple parity sequence BHD schemes have more than one channel state sequence, and therefore one computation is slightly more expensive than in the single parity sequence case.
- Turbo-codes, though, perform the same number of computations per bit: 3 multiplications per state times 16 states times 18 iterations, for a total of 864 computations per decoded bit, far exceeding the computational requirements of BHD at SNR's above 1.2dB.


Bit Error Rate for BHD with 1,2,3 Parity Sequences
Bit Error Rate
10^-2
10^-3
10^-4
10^-5
10^-6
0.0
1.0
2.0
E_b/N_0 (dB)
Capacity
Turbo-code
BHD/2
BHD/1
BHD/3
Cutoff Rate



Number of Computations for BHD with 1,2,3 Parity Sequences
1000
100
10
Number of Computations
0.0
1.0
2.0
E_b/N_0 (dB)
BHD/3
BHD/2
BHD/1

Conclusions

- An extension of the Bootstrap Hybrid Decoding scheme to multiple parity sequences was presented, detailing how to obtain the metric adjustments for the sequential decoders, as well as the channel state stream updates based on successful decoder estimates.
- Simulation results indicate that such schemes can add 0.2dB to the original scheme's performance, thus coming closer to channel capacity, at a cost of some increased decoding complexity and delay.
- Analysis of the computational behavior is being performed to determine the theoretical limits of this scheme.